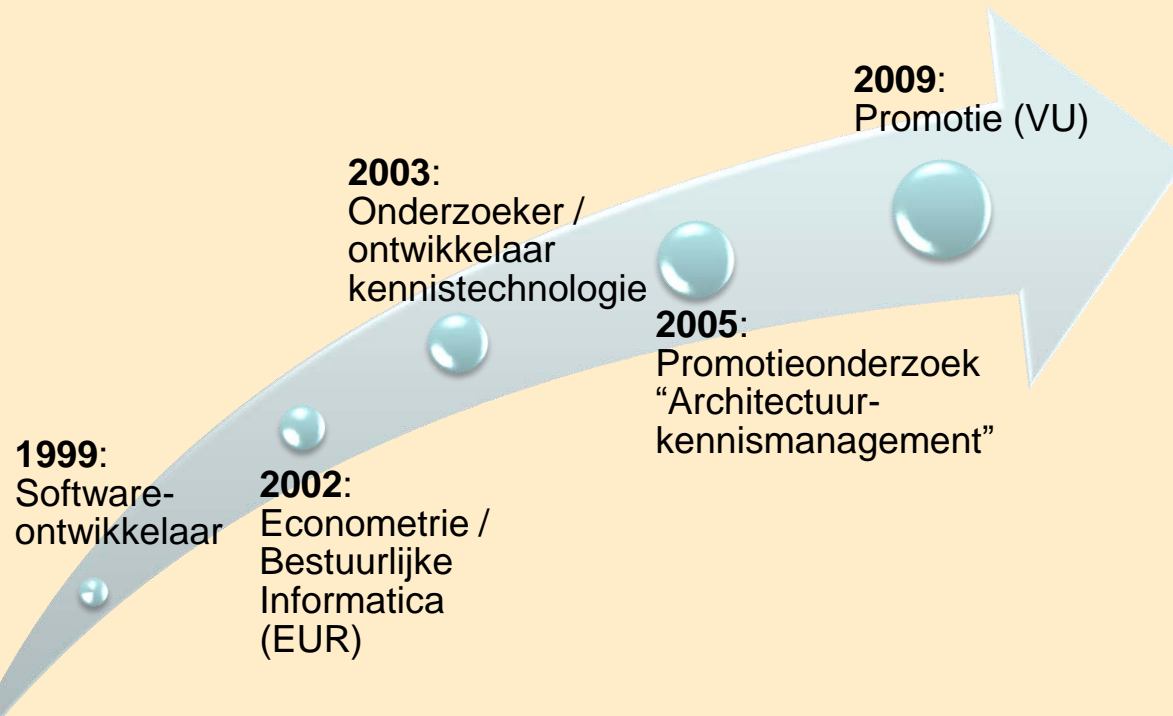




# **NFR & Architectuur: Twee handen op één buik**

Remco de Boer

# Over mijzelf



**Remco de Boer**  
*Consultant*

[rdeboer@archixl.nl](mailto:rdeboer@archixl.nl)  
+31 (0)6 2426 84 35



**ArchiXL B.V.**  
Nijverheidsweg Noord 60-27  
3812 PM Amersfoort

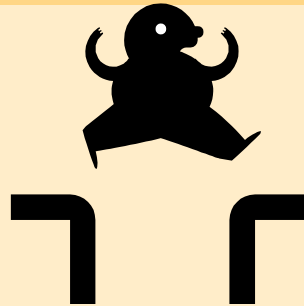
# Agenda

- Requirements vs. architectuur
- Requirements engineering als probleemanalyse
- Architectuur: van 'structuregeoriënteerd' naar 'kennisgeoriënteerd'
- Requirements vs. architectuurontwerpbeslissingen
- Conclusie en aanbevelingen

# Requirements

vs.

# Architectuur



Wat?  $\longleftrightarrow$  Hoe?

Probleem  $\longleftrightarrow$  Oplossing

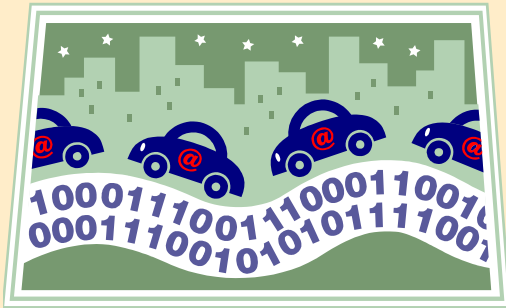
Vóór ondertekening  $\longleftrightarrow$  Na ondertekening

Swartout & Waltzer (1982):  
*On the inevitable intertwining of  
specification and implementation*

“Staat vast”  $\longleftrightarrow$  “Nog te bepalen”

# Requirements engineering als problemanalyse

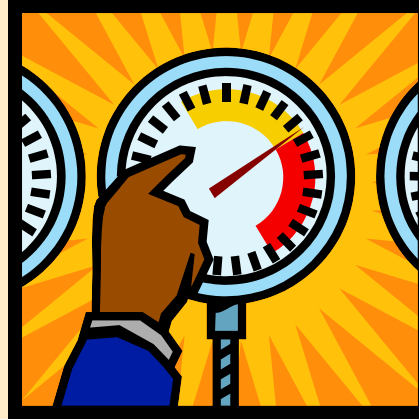
'Indicatief'



## Problem domain

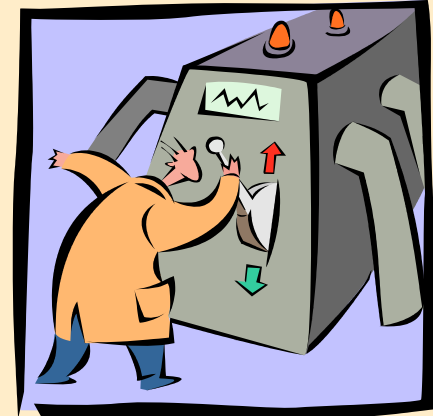
Het deel van de wereld waar het probleem zich bevindt

'Optatief'



## Requirements

Beschrijving van wat de klant waar zou willen laten zijn in de probleemwereld



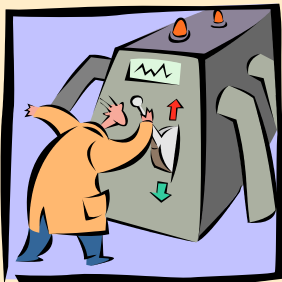
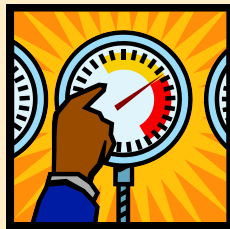
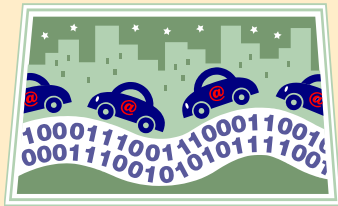
## Machine specification

Beschrijving van het gewenste gedrag van de machine in de probleemwereld  
(*black box*)

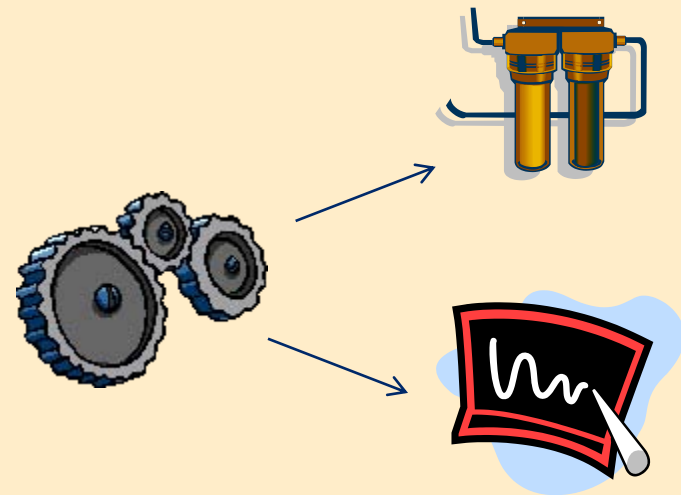
*Jackson (2001): Problem frames*

# Probleem en oplossing staan niet los van elkaar

Probleem



Oplossing



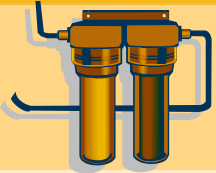
## Voorbeeld: KWIC

- KWIC: “Key Word In Context”
  - Hans Peter Luhn (IBM, 1958)
  - Parnas (1972): *On the criteria to be used in decomposing systems into modules*
- Werking van een KWIC index:
  - ‘circular shifting’ van een regel tekst (bijv. titel van een publicatie)
    - haal het eerste woord weg
    - plak dit aan het eind van de regel
  - resultaat: lexicografisch geordende lijst van alle gepermuteerde regels
- Circular shifting: Software Architecture in Practice (Bass e.a., 2003)
  - Software Architecture in Practice
  - Architecture in Practice / Software
  - in Practice / Software Architecture
  - Practice / Software Architecture in

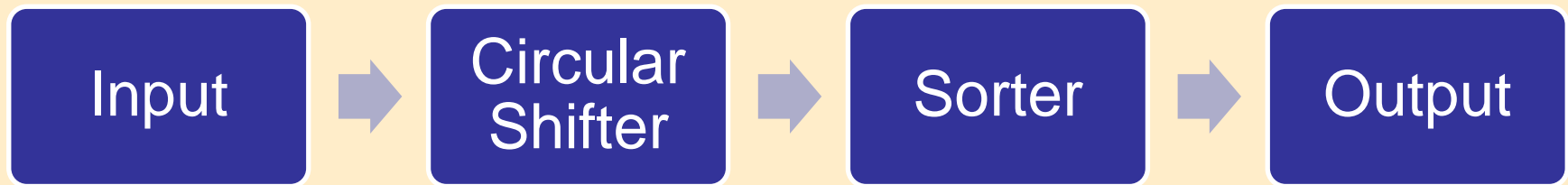
# KWIC Index

Context	Key Word	Context	<i>Publicatie</i>
Software	<b>Architecture</b>	in Practice	Bass et al. (2003)
Design & Use of Software	<b>Architectures</b>		Bosch (2000)
	<b>Design</b>	& Use of Software Architectures	Bosch (2000)
Software	<b>Engineering</b>	Principles and Practice	Van Vliet (2009)
Software Architecture in	<b>Practice</b>		Bass et al. (2003)
Software Engineering Principles and	<b>Practice</b>		Van Vliet (2009)
Software Engineering	<b>Principles</b>	and Practice	Van Vliet (2009)
	<b>Software</b>	Architecture in Practice	Bass et al. (2003)
Design & Use of	<b>Software</b>	Architectures	Bosch (2000)
	<b>Software</b>	Engineering Principles And Practice	Van Vliet (2009)
Design &	<b>Use</b>	of Software Architectures	Bosch (2000)



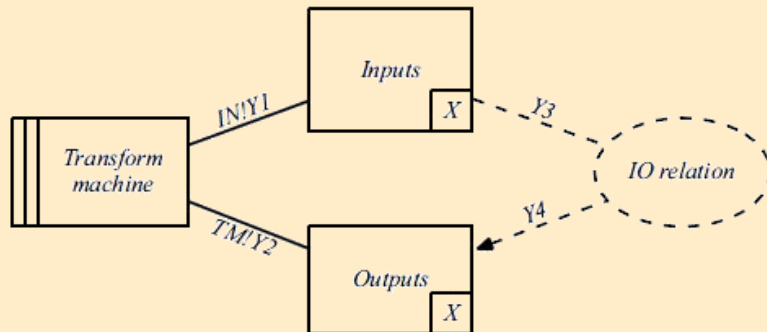


## Pipe-and-filter architectuur voor KWIC



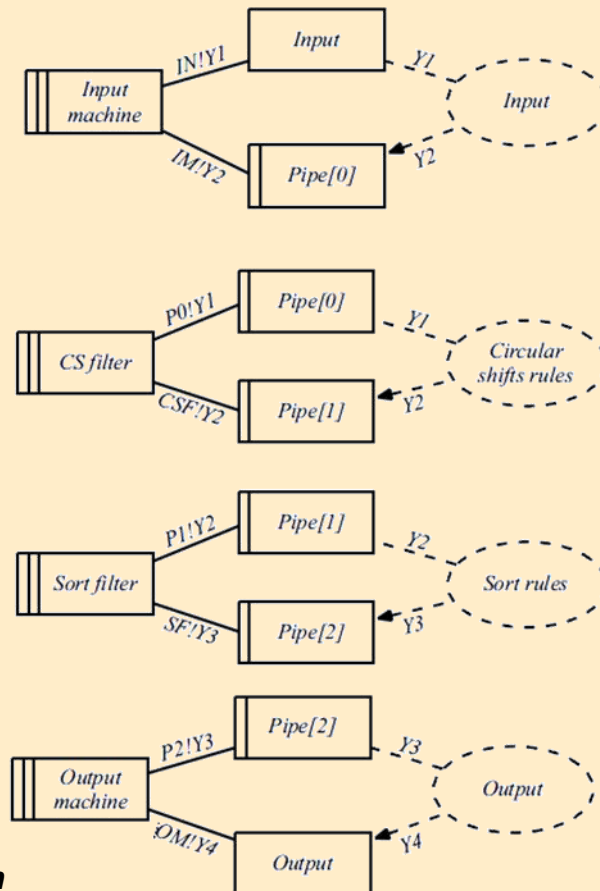
# Probleemanalyse voor en na architectuurkeuze

## KWIC

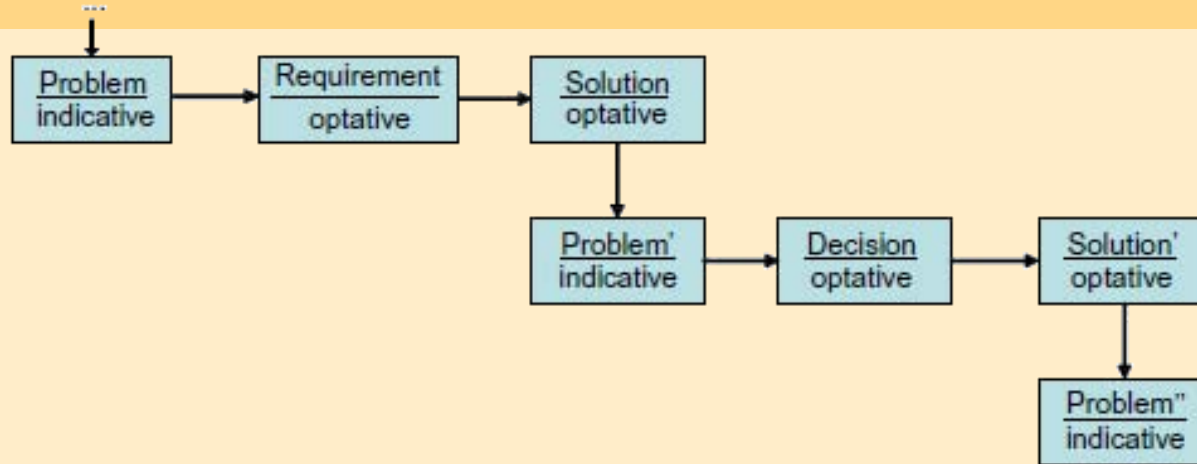


## KWIC:

### Pipe & Filter architectuur



# Architectuurkeuze verandert de probleemwereld

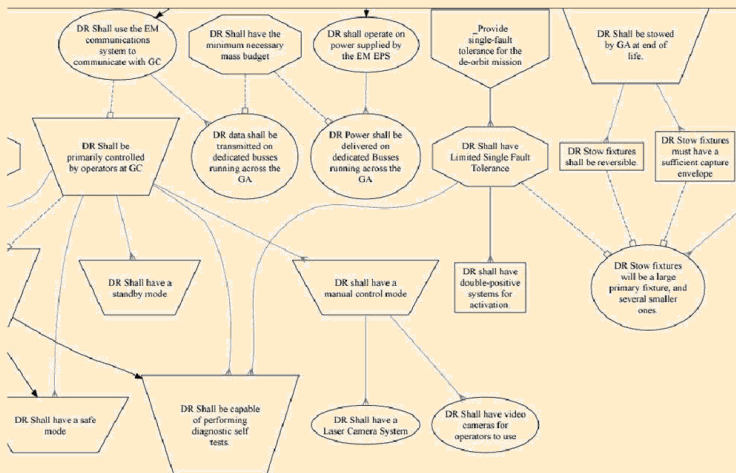


- Gebruik van pipes & filters
  - is een keuze (optatief), maar
  - wordt vervolgens een gegeven (indicatief)
- NB. Pipe&filter is maar één van de mogelijke architectuurkeuzen
  - mogelijke alternatieven: shared data, abstract data type, implicit invocation
  - Geschiktheid hangt mede af van de gestelde (niet-functionele) eisen *en hun relatieve onderlinge prioriteit*.

(cf. Chung et al. (1999), *Architectural Design to Meet Stakeholder Requirements*)

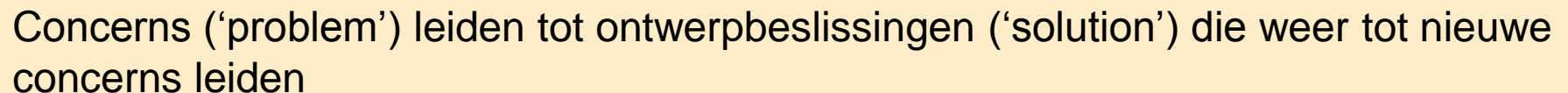
# Architectuur is meer dan structuur

- Van 'structuurgeoriënteerd' naar 'kennisgeoriënteerd'
- Architectuur = structuur + ontwerpbeslissingen



Decision graph (Kruchten et al., 2005))

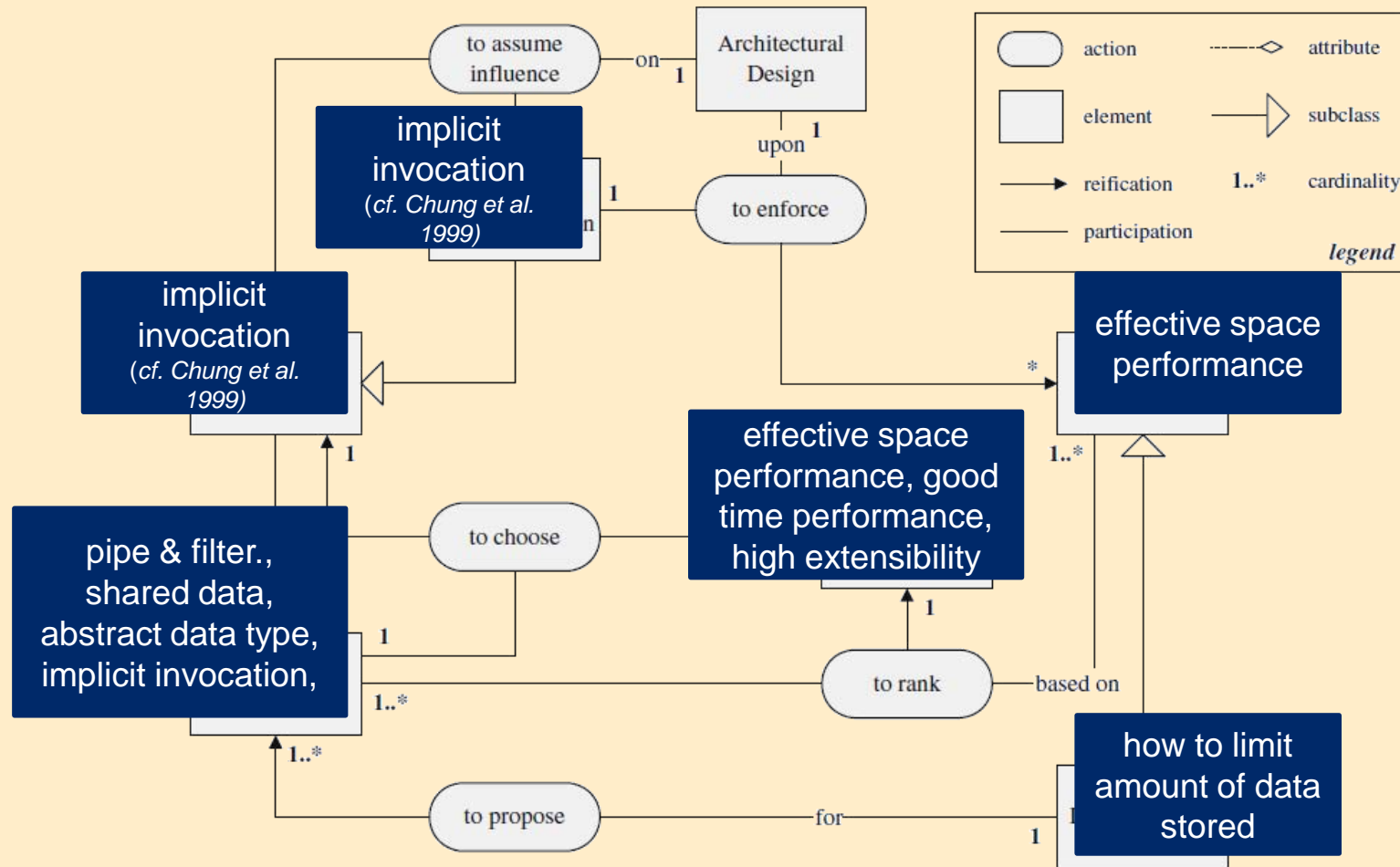




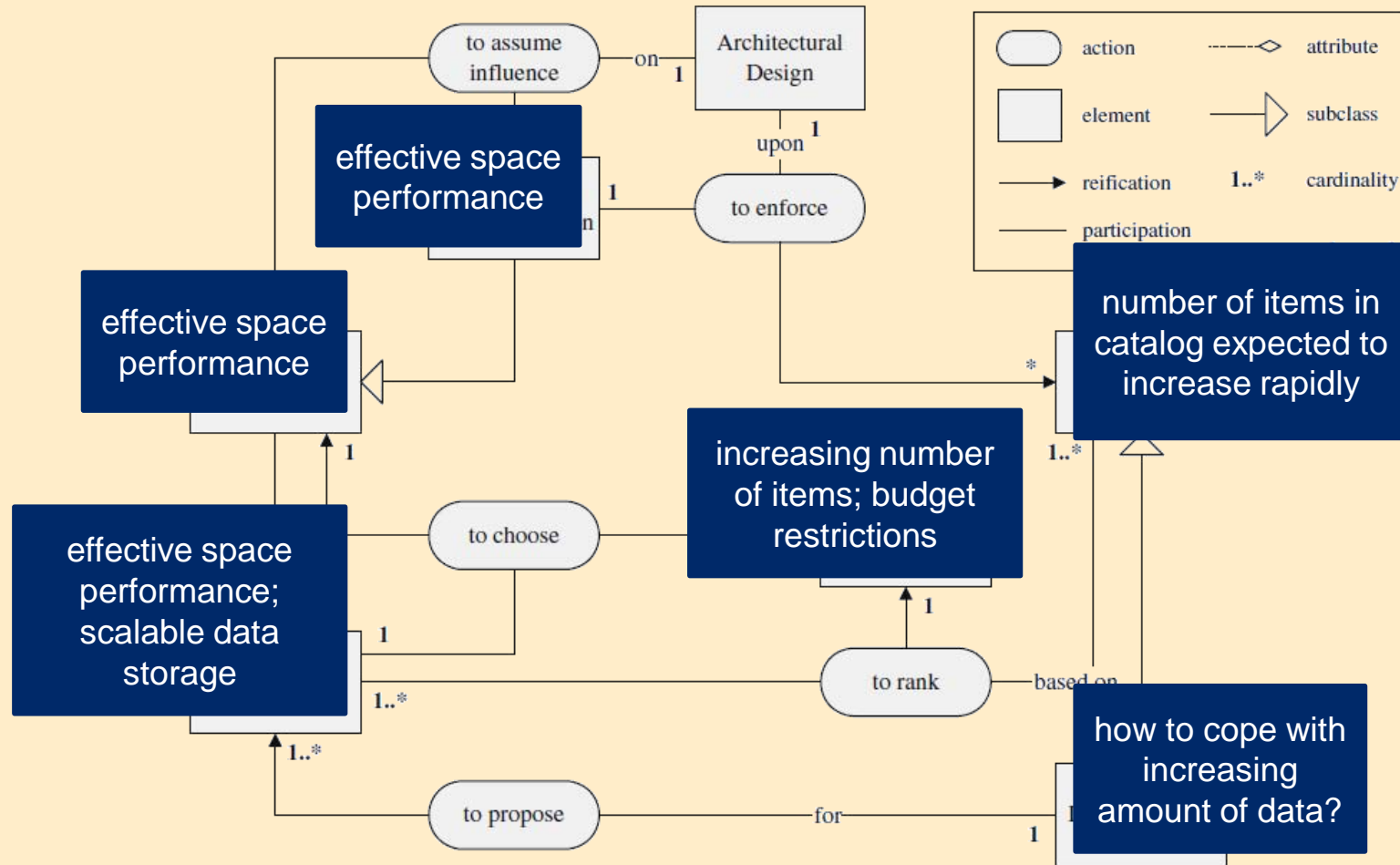
## Intermezzo: requirement of ontwerpbeslissing?

***“The KWIC component must have effective space performance”***

# Requirements vs. ontwerpbeslissingen: Nogmaals het KWIC probleem

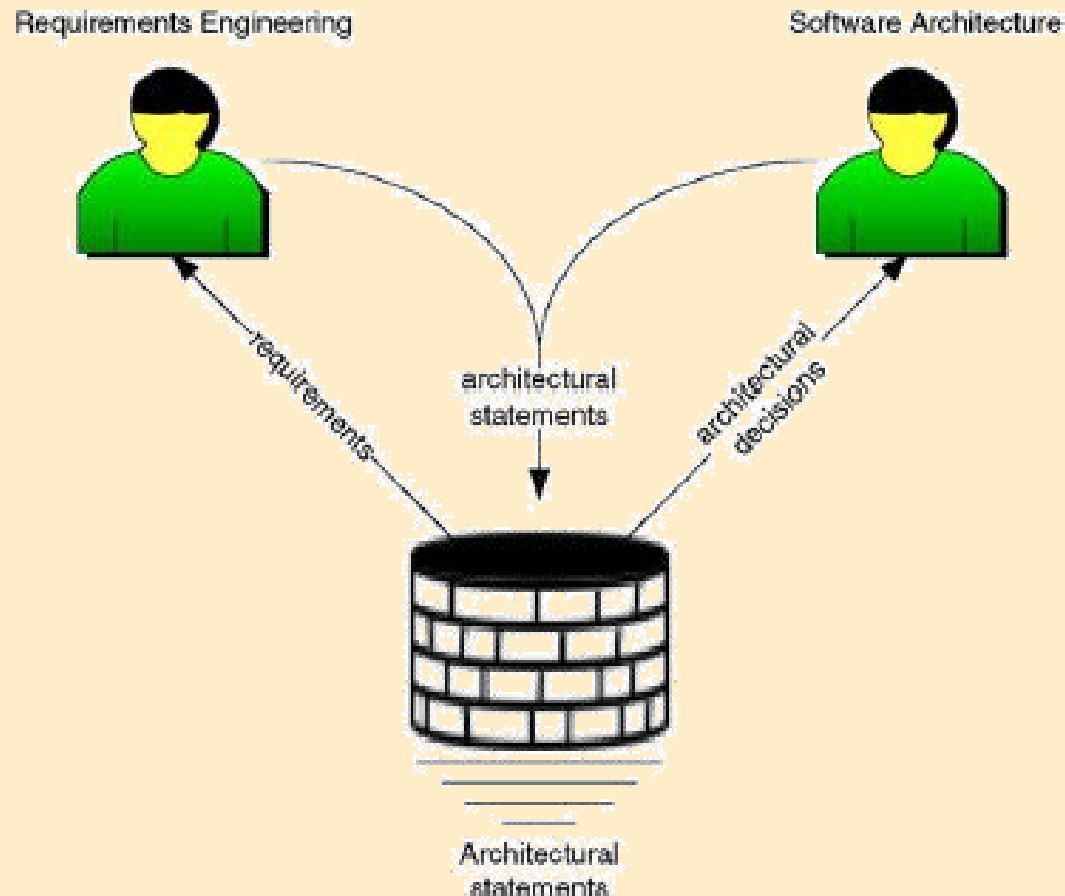


# Requirements vs. ontwerpbeslissingen: Nogmaals het KWIC probleem





# Requirements en architectuur in de wensput van architectuuruitspraken



# Methoden en technieken rondom de wensput

Requirements engineering	<i>de 'wensput'</i>	Architectuur
Elicitation	<i>creation of statements</i>	Decision making
Negotiation		Trade-off analyses
Specification	<i>drop statements in the well</i>	Design
Validation	<i>compare well contents with reality</i>	Assessment
Documentation	<i>write down well contents</i>	Description
Requirements management	<i>structure well contents</i>	Architectural knowledge management

# Conclusie



- Er is geen fundamenteel verschil tussen requirements en ontwerpbeslissingen
  - Beide zijn een vorm van architectuuruitspraken die het uiteindelijke ontwerp beïnvloeden
  - Onderscheid is arbitrair (maar kan wel nuttig zijn in een gegeven context)
- Aanbevelingen:
  - Maak onderscheid waar dat nuttig is
    - Bijv. formele verantwoordelijkheden (voordat/nadat het contract is ondertekend)
  - Wees je bewust van de wederzijdse afhankelijkheden
    - De architect heeft een rol in het requirementsproces
  - Maak gebruik van de overeenkomsten!
    - Kennismanagement
    - Traceerbaarheid
    - Methoden en technieken